

---

# A solution to the model attribution challenge<sup>\*</sup>

---

Hieu Minh Nguyen<sup>†</sup>  
jordnguyen43@gmail.com

## Abstract

This paper introduces a methodology for solving the problem of attributing fine-tuned models to their corresponding open-source base models, with limited access only through a black-box API. The proposed solution combines human evaluation of the model’s training data, analysis of output similarity compared to the base models, and observations regarding API behaviour. Various attribution approaches are employed, including domain analysis, language characteristics, output token length, identification of gibberish or repetition, identification of same continuations, and analysis of API loading time and failures. The methodology was developed during the Machine Learning Model Attribution Challenge, which was organised as part of the 1st IEEE Conference on Secure and Trustworthy Machine Learning. The solution secured third place, successfully attributing 6 out of 12 models by performing 244 API queries.

*Index terms:* fine-tuned models, base models, model attribution, black-box access, API, Machine Learning Model Attribution Challenge.

## 1 Introduction

The past decade has witnessed remarkable advancements in the development of Large Language Models (LLMs), enabling machines to tackle complex real-world problems that were previously unimaginable. With the release of open-source foundation models, such as GPT, the boundaries between reality and artificial simulation have become increasingly blurred. While these LLMs have revolutionised various domains, they have also raised concerns regarding their potential for malicious use when made more accessible, affordable, and easier to manipulate.

Open-sourcing foundation models has paved the way for fine-tuning, allowing users to customise LLMs for specific tasks or domains. However, this accessibility comes with risks, as it becomes easier for individuals to exploit LLMs for nefarious purposes. Malicious actors can fine-tune these models on attractive domain-specific topics, potentially leading to misinformation, propaganda, or harmful content generation. This makes it crucial to develop mechanisms that can trace the origins of synthetic text generated by fine-tuned models and attribute them back to the base models. Without a generalised forensic process to trace textual output, it becomes challenging to regulate, track, and remediate instances where LLMs are misused.

To address this critical issue, the Machine Learning Model Attribution Challenge (MLMAC) was established. The competition aims to explore what traces of provenance can be gleaned from synthetic text that lacks explicit marking. The ultimate goal is to develop attribution techniques that serve as powerful tools for regulation, tracking, and remediation when LLMs are misused. Currently, there is no established method to trace the textual output from customised models back to the base models, making this challenge even more significant.

In the MLMAC, contestants are tasked with developing creative solutions to uncover the provenance of models. Contestants interact with a set of fine-tuned models through a text generation API,

---

<sup>\*</sup>Research done during the MLMAC

<sup>†</sup>github

attributing each generated output back to a known set of LLMs, referred to as "base models." By building forensic capabilities and establishing the difficulty of model attribution, this challenge takes a crucial step toward ensuring the responsible and trustworthy use of LLMs and artificial intelligence in general.

The competition presents two sets of models: a set of twelve base models and a set of twelve fine-tuned models. Contestants' solutions will be evaluated based on the correctness of their submitted results, specifically the number of correct (fine-tuned model, base model) pairs identified. In case of ties, the number of API queries used to interact with the anonymous fine-tuned models will be considered, with fewer queries being advantageous. Additionally, the submission time will be taken into account, with earlier submissions breaking any subsequent ties.

## 2 Methodology

The proposed methodology for model attribution in the context of fine-tuned models and their associated base models involves a systematic approach to establish connections and identify the origins of the fine-tuned models. The methodology encompasses the development of heuristics, analysis of publicly available information, and comparison of model characteristics and responses.

To begin with, a set of heuristics is devised to facilitate the attribution process. These heuristics are designed to identify patterns, similarities, and distinguishing features between the fine-tuned models and their respective base models. These criteria serve as guidelines for connecting the models based on various factors and evidence.

Publicly accessible information, primarily obtained from platforms such as Hugging Face, plays a crucial role in the attribution process. This information includes pertinent details about the base models, such as the origin of the training data, the size of the models, and the scope of the training data. Understanding the training data's nature (e.g., multilingual or English-focused) and composition (e.g., inclusion of code, corpus of books, or social media data) aids in establishing associations and identifying potential base models.

Prompting the base models with diverse queries is an integral part of the methodology. These queries encompass a wide range of categories, including zero-shot prompts, one-shot prompts, sentimental analysis, coding questions, arbitrary gibberish, and factual inquiries. By studying the responses generated by the base models, distinctive characteristics and idiosyncrasies are discerned, enabling subsequent comparisons.

Model characteristics, encompassing various aspects such as the nature of the training data and the intended model type (e.g., long-form text generation or conversational chatbot), are meticulously examined and documented. These characteristics serve as reference points for subsequent analyses and serve to differentiate base models from one another.

To establish connections between the fine-tuned models and the base models, the fine-tuned models are prompted with similar queries. By assessing and comparing the responses of the fine-tuned models with those of the base models, commonalities, patterns, and similarities are identified. This comparative analysis aids in attributing the fine-tuned models to their respective base models, providing insights into their provenance.

It is essential to acknowledge that multiple fine-tuned models may be associated with a single base model, or a fine-tuned model may be linked to a base model that was not provided in the competition dataset. Therefore, it is crucial to exercise caution when relying solely on the most similar output as it does not guarantee exclusive attribution to a specific base model.

By following this comprehensive methodology, the proposed solution aims to establish robust and reliable model attribution techniques, even in scenarios where explicit information or detailed model architecture is not available. Through systematic analysis and comparison, this methodology provides valuable insights for accurately attributing fine-tuned models to their base models.

Keywords: model attribution, fine-tuned models, base models, heuristics, publicly available information, model characteristics, querying, pattern identification, comparative analysis, provenance.

Table 1: Provided base models in the challenge

No.	Base model
1	bloom-2b5
2	bloom-350m
3	Multilingual-MiniLM-L12-H384
4	distilgpt2
5	codegen-350M-multi
6	opt-350m
7	gpt2-xl
8	gpt-neo-125M
9	xlnet-base-cased
10	DialoGPT-large
11	gpt2
12	gpt-j-6B

### 3 Experimentation with base models

Open-source base models were provided to the contestants before the starting date.

Experiments looked like this. The model completes the text provided by the user.

User: Once upon a time, there was

Model: Once upon a time, there was a great kingdom that spanned continents. One day, the king of said kingdom decided

#### 3.1 Domain of training data

Base models might have been trained for a specific purpose (e.g writing code), thus their training data is likely to come from a corpus mainly composed of text related to such purpose. For example, Codegen-350M-multi was trained on a large-scale dataset of multiple programming languages from GitHub repositories. The data consists of 119.2B tokens and includes C, C++, Go, Java, JavaScript, and Python.

As a result, almost everything outputted by this model ends up related to programming. In this particular case, the model outputs something akin to a programming tutorial script.

User: Once upon a time,

Model: Once upon a time, this is the main loop that would be used. If you're going to use the main loop for some other purpose, just do "run()" instead of this. "run()" will have its own event loop if and I am learning in Java and I want to learn how do know that we will be using Java, I create a client that uses the Java client API. With this server you create a client for

#### 3.2 Multilingual coherency

In a similar vein, as data scraped from the Internet is predominantly English (cite), some models are specifically trained on a multilingual dataset to serve purposes.

For instance, the two BLOOM models were both trained on multiple languages. They achieve much greater coherency on non-English prompts than other base models.

A prompt in Vietnamese

User: phở là món ăn

Model: phở là món ăn được nhiều người yêu thích. Tuy nhiên, không phải ai cũng biết cách

Translated:

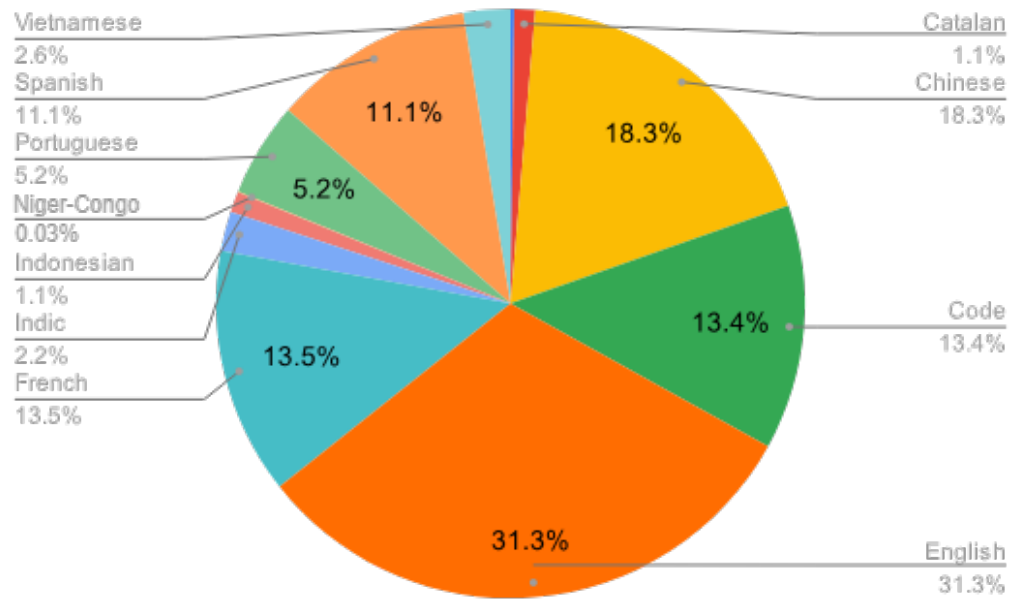


Figure 1: BLOOM's training data consists of multiple languages

User: pho is a dish

Model: pho is a dish **that is favoured by many. However, not everyone knows**

### 3.3 Output length and repetition

Another detectable variation between models is their output length. In particular, xl-net-base-cased is a model trained to achieve high performance on long form text generation.

User: the best way to

Model: the best way to **get in the door is by ordering the right kind of food and a proper diet... When no one bothers to get in the door, you want to eat a healthy lunch or dinner. It will be important to have a big lunch with plenty of calories in the form of a decent meal and good water. Once you are out, try to have a nice breakfast of something to go with your lunch. The good thing about eating a healthy breakfast you can never have a bad lunch. In addition, if you do have a bad lunch, make sure that you eat it. The good things about eating healthy are that you can never have a bad lunch and that you don't have to pay a lot of money to eat. The good thing about eating healthy is that you don't have to pay a lot of money, that you don't have to spend a lot of money on food and that you don't have to do a lot of shopping, that you don't have to spend a lot of money on car (that you can go shopping from anywhere you want, that you don't have to pay too much money for fuel), that you don't have to pay too**

A special characteristics of xl-net is consistently generating long and nonsensical repetitions.

### 3.4 Model size and API observations

Some models can take significantly more time to respond through the API than other models. A highly probable cause of this is the huge disparity between model sizes, indicated by their number of trainable parameters. This is explained by the fact that some models are trained for a wide variety of tasks while some are made specifically as distilled versions. Additionally, since all of the provided models were hosted on huggingface, we can rule out internet connection speed and stability is ruled out as possible sources.

Some large models such as gpt-j-6B (see table 2), can take significantly more time to respond, up to several minutes.

Table 2: Number of parameters in base models, in millions of parameters

Model	Parameter Size
bloom-2b5	2500
bloom-350m	350
Multilingual-MiniLM-L12-H384	21
distilgpt2	82
codegen-350M-multi	350
opt-350m	350
gpt2-xl	1558
gpt-neo-125M	125
xlnet-base-cased	110
DialoGPT-large	774
gpt2	117
gpt-j-6B	6053

### 3.5 Sentimental analysis and unique continuations

Sentimental analysis is a common technique used to measure whether the model understands the overall idea behind a sentence or paragraph.

I picked a reasonably long Steam review from Hollow Knight detailing a particularly difficult section of the game. I then appended a question to the end of the paragraph to ask the model what the reviewer thinks of the game.

User: I agree with the criticism of WP, however. That place had me tearing my hair out.

Aside from being a pretty sudden departure from most of the rest of the game, it's just annoying to deal with. It relies on an infuriating trial and error system in some areas, and is just plain frustrating in others. The fact that its mandatory for two of the three endings makes it even worse.

It's not bad enough to make the game bad, but it's boring, bland and frustrating low point of the game.

The review thinks that the game is

Model: I agree with the criticism of WP, however. That place had me tearing my hair out.

Aside from being a pretty sudden departure from most of the rest of the game, it's just annoying to deal with. It relies on an infuriating trial and error system in some areas, and is just plain frustrating in others. The fact that its mandatory for two of the three endings makes it even worse.

It's not bad enough to make the game bad, but it's boring, bland and frustrating low point of the game.

The review thinks that the game is **boring**

The review was nuanced enough (it just says the game is not bad) to make all of the models output incorrectly, but there were some interesting results.

For example, every single model in the GPT family outputted "icky".

Table 3: Final submission

Fine-tuned model.	Base model
0	DialoGPT-large
1	bloom-2b5
2	bloom-2b5
3	Multilingual-MiniLM-L12-H384
4	gpt-xl
5	codegen-350M-multi
6	opt-350m
7	codegen-350M-multi
8	gpt-neo-125M
9	gpt-j-6B
10	xlnet-base-cased
11	gpt2

### 3.6 One-shot prompts

As opposed to "zero-shot" prompts where the model is asked to complete sentences with zero examples (see all of the previous examples), "one-shot" prompts provide the model with an example for its response. This can improve model performance, and steer the model in favour of more coherent outputs.

User: Millenium math problems include: 1.The Riemann Hypothesis  
Model: Millenium math problems include: 1.The Riemann Hypothesis 2. Inverse Euler's Formula 3. Hypergeometric Sequences 4. Hahn-Banach Geometry 5. Hyper

This example comes from gpt-xl. Smaller models like gpt-neo or opt only returned "2."

## 4 Experimentation with fine-tuned models, deduction, and attribution

Will go through each model in order of presented methods

12 fine-tuned models

Based on this method, here is the final submitted attribution.

### 4.1 Model 7

codegen

### 4.2 Model 10

xl net

### 4.3 Model 9

gpt-j-6b

### 4.4 Model 2

bloom 2b5

### 4.5 Model 4

model 3 sometimes generates unprompted introductions. My guess was dialogpt-large.

model 4 (gptxl) was finetuned on r/writingprompts ([WP] and reddit usernames).

model 6 was finetuned on Crime and Punishment by Dostoyevsky (character names and passages consistently).

model 8 was finetuned on some The Beatles music lyrics, which I picked gpt-neo for short lines and similarities to other gpt models.

## 5 Conclusion and discussion

This method got 50% accuracy, getting 6/12 models correct.

According to the MLMAC report paper, the most successful approaches by other participants is in a similar vein, of evaluating the similarity between model outputs or through observations on the metadata (model size, training data characteristics)

Some other methods such as temporal range,

temporal range cutoff date and model specific special tokens [the blog post with the diagram, cite and include it],

It is important to note that the competition stated that this is a naive adversary. In realistic scenarios they may finetune the model on training data with dates later then the original training, or intentionally avoid the mention of anything related to the provenance (through RLHF, LLM filters).

And with the sheer volume of possible adversaries, manual evaluation is not a realistic solution to model attribution. Some participants have had some success with automated methods [cite 4th place guy]

Cite the other participant's arxiv paper

## 6 Acknowledgements

I would like to thank MITRE, Microsoft, Schmidt-Futures, Robust-Intelligence, Lincoln-Network, and the Huggingface community for hosting the Machine Learning Model Attribution Challenge. I would also like to thank Intern Paul for their advice on this paper.

### 6.1 Style

Papers to be submitted to NeurIPS 2023 must be prepared according to the instructions presented here. Papers may only be up to **five** pages long, including figures. Additional pages *containing only acknowledgments and references* are allowed. Papers that exceed the page limit will not be reviewed, or in any other way considered for presentation at the conference.

The margins in 2023 are the same as those in 2007, which allow for  $\sim 15\%$  more words in the paper compared to earlier years.

Authors are required to use the NeurIPS L<sup>A</sup>T<sub>E</sub>X style files obtainable at the NeurIPS website as indicated below. Please make sure you use the current files and not previous versions. Tweaking the style files may be grounds for rejection.

### 6.2 Retrieval of style files

The style files for NeurIPS and other conference information are available on the World Wide Web at

<http://www.neurips.cc/>

The file `neurips_2023.pdf` contains these instructions and illustrates the various formatting requirements your NeurIPS paper must satisfy.

The only supported style file for NeurIPS 2023 is `neurips_2023.sty`, rewritten for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. **Previous style files for L<sup>A</sup>T<sub>E</sub>X 2.09, Microsoft Word, and RTF are no longer supported!**

The L<sup>A</sup>T<sub>E</sub>X style file contains three optional arguments: `final`, which creates a camera-ready copy, `preprint`, which creates a preprint for submission to, e.g., arXiv, and `nonatbib`, which will not load the `natbib` package for you in case of package clash.

**Preprint option** If you wish to post a preprint of your work online, e.g., on arXiv, using the NeurIPS style, please use the `preprint` option. This will create a nonanonymized version of your work with the text “Preprint. Work in progress.” in the footer. This version may be distributed as you see fit. Please **do not** use the `final` option, which should **only** be used for papers accepted to NeurIPS.

At submission time, please omit the `final` and `preprint` options. This will anonymize your submission and add line numbers to aid review. Please do *not* refer to these line numbers in your paper as they will be removed during generation of camera-ready copies.

The file `neurips_2023.tex` may be used as a “shell” for writing your paper. All you have to do is replace the author, title, abstract, and text of the paper with your own.

The formatting instructions contained in these style files are summarized in Sections 7, 8, and 9 below.

## 7 Base model prompting

### 7.1 Training data

### 7.2 Model architecture and parameter size

The text must be confined within a rectangle 5.5 inches (33 picas) wide and 9 inches (54 picas) long. The left margin is 1.5 inch (9 picas). Use 10 point type with a vertical spacing (leading) of 11 points. Times New Roman is the preferred typeface throughout, and will be selected for you by default. Paragraphs are separated by 1/2 line space (5.5 points), with no indentation.

The paper title should be 17 point, initial caps/lower case, bold, centered between two horizontal rules. The top rule should be 4 points thick and the bottom rule should be 1 point thick. Allow 1/4 inch space above and below the title to rules. All pages should start at 1 inch (6 picas) from the top of the page.

For the final version, authors’ names are set in boldface, and each name is centered above the corresponding address. The lead author’s name is to be listed first (left-most), and the co-authors’ names (if different address) are set to follow. If there is only one co-author, list both author and co-author side by side.

Please pay special attention to the instructions in Section 9 regarding figures, tables, acknowledgments, and references.

## 8 Fine-tuned model prompting

All headings should be lower case (except for first word and proper nouns), flush left, and bold.

First-level headings should be in 12-point type.

### 8.1 Headings: second level

Second-level headings should be in 10-point type.

#### 8.1.1 Headings: third level

Third-level headings should be in 10-point type.

**Paragraphs** There is also a `\paragraph` command available, which sets the heading in bold, flush left, and inline with the text, with the heading followed by 1 em of space.



## 9 Citations, figures, tables, references

These instructions apply to everyone.

### 9.1 Citations within the text

The `natbib` package will be loaded for you by default. Citations may be author/year or numeric, as long as you maintain internal consistency. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

The documentation for `natbib` may be found at

<http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf>

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dotso
```

produces

Hasselmo, et al. (1995) investigated...

If you wish to load the `natbib` package with options, you may add the following before loading the `neurips_2023` package:

```
\PassOptionsToPackage{options}{natbib}
```

If `natbib` clashes with another package you load, you can add the optional argument `nonatbib` when loading the style file:

```
\usepackage[nonatbib]{neurips_2023}
```

As submission is double blind, refer to your own published work in the third person. That is, use “In the previous work of Jones et al. [4],” not “In our previous work [4].” If you cite your other papers that are not widely available (e.g., a journal paper under review), use anonymous author names in the citation, e.g., an author of the form “A. Anonymous.”

### 9.2 Footnotes

Footnotes should be used sparingly. If you do require a footnote, indicate footnotes with a number<sup>3</sup> in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).

Note that footnotes are properly typeset *after* punctuation marks.<sup>4</sup>

### 9.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction. The figure number and caption always appear after the figure. Place one line space before the figure caption and one line space after the figure. The figure caption should be lower case (except for first word and proper nouns); figures are numbered consecutively.

You may use color figures. However, it is best for the figure captions and the paper body to be legible if the paper is printed in either black/white or in color.

### 9.4 Tables

All tables must be centered, neat, clean and legible. The table number and title always appear before the table. See Table 4.

---

<sup>3</sup>Sample of the first footnote.

<sup>4</sup>As in this example.

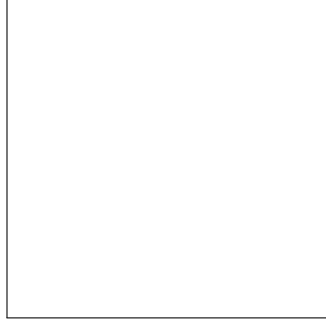


Figure 2: Sample figure caption.

Table 4: Sample table title

Part		
Name	Description	Size ( $\mu\text{m}$ )
Dendrite	Input terminal	$\sim 100$
Axon	Output terminal	$\sim 10$
Soma	Cell body	up to $10^6$

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

Note that publication-quality tables *do not contain vertical rules*. We strongly suggest the use of the booktabs package, which allows for typesetting high-quality, professional tables:

<https://www.ctan.org/pkg/booktabs>

This package was used to typeset Table 4.

## 10 Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

## 11 Preparing PDF files

Please prepare submission files with paper size “US Letter,” and not, for example, “A4.”

Fonts were the main cause of problems in the past years. Your PDF file must only contain Type 1 or Embedded TrueType fonts. Here are a few instructions to achieve this.

- You should directly generate PDF files using `pdflatex`.
- You can check which fonts a PDF files uses. In Acrobat Reader, select the menu Files>Document Properties>Fonts and select Show All Fonts. You can also use the program `pdf fonts` which comes with `xpdf` and is available out-of-the-box on most Linux machines.
- The IEEE has recommendations for generating PDF files whose fonts are also acceptable for NeurIPS. Please see <http://www.emfield.org/icuwb2010/downloads/IEEE-PDF-SpecV32.pdf>
- `xfig` "patterned" shapes are implemented with bitmap fonts. Use "solid" shapes instead.
- The `\bbold` package almost always uses bitmap fonts. You should use the equivalent AMS Fonts:

```
\usepackage{amsfonts}
```

followed by, e.g., `\mathbb{R}`, `\mathbb{N}`, or `\mathbb{C}` for  $\mathbb{R}$ ,  $\mathbb{N}$  or  $\mathbb{C}$ . You can also use the following workaround for reals, natural and complex:

```
\newcommand{\RR}{\mathbb{R}} %real numbers
\newcommand{\Nat}{\mathbb{N}} %natural numbers
\newcommand{\CC}{\mathbb{C}} %complex numbers
```

Note that `amsfonts` is automatically loaded by the `amssymb` package.

If your file contains type 3 fonts or non embedded TrueType fonts, we will ask you to fix it.

## 11.1 Margins in L<sup>A</sup>T<sub>E</sub>X

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the `graphicx` package. Always specify the figure width as a multiple of the line width as in the example below:

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

See Section 4.4 in the graphics bundle documentation (<http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf>)

A number of width problems arise when L<sup>A</sup>T<sub>E</sub>X cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command when necessary.

## Acknowledgments and Disclosure of Funding

Use unnumbered first level headings for the acknowledgments. All acknowledgments go at the end of the paper before the list of references. Moreover, you are required to declare funding (financial activities supporting the submitted work) and competing interests (related financial activities outside the submitted work). More information about this disclosure can be found at: <https://neurips.cc/Conferences/2023/PaperInformation/FundingDisclosure>.

Do **not** include this section in the anonymized submission, only in the final paper. You can use the `ack` environment provided in the style file to automatically hide this section in the anonymized submission.

## References

References follow the acknowledgments. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. Note that the Reference section does not count towards the page limit.

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems* 7, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural Simulation System*. New York: TELOS/Springer-Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

Here is a bunch of experiments with the language models